# Getting Started

**Your Fast Track to Solutions**

Microsoft, Windows, Windows NT, Windows 2000, Windows XP, Visual Basic, Microsoft .NET, Visual C++, Visual C#, and ActiveX are either trademarks or registered trademarks of Microsoft Corporation.

All other nationally and internationally recognized trademarks and tradenames are hereby recognized.

Copyright © 2000-2008 by MVTec Software GmbH, München, Germany

| | | |
|---|---|---|
| Edition 1 | November 2000 | (ActivVisionTools 1.0) |
| Edition 2 | Februar 2001 | (ActivVisionTools 1.2) |
| Edition 3 | April 2001 | (ActivVisionTools 1.3) |
| Edition 4 | September 2001 | (ActivVisionTools 2.0) |
| Edition 5 | November 2002 | (ActivVisionTools 2.1) |
| Edition 6 | May 2003 | (ActivVisionTools 2.2) |
| Edition 7 | January 2005 | (ActivVisionTools 3.0) |
| Edition 8 | February 2006 | (ActivVisionTools 3.1) |
| Edition 9 | May 2008 | (ActivVisionTools 3.2) |

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of the publisher.

More information about ActivVisionTools can be found at:

`http://www.activ-vision-tools.com`

# How to Read This Manual

This manual is a quick guide to ActivVisionTools, the family of tools for the world of machine vision. It provides all necessary information to install and configure ActivVisionTools and to understand their basic philosophy.

A first example session shows how to work with ActivVisionTools within the framework of Microsoft Visual Basic.

# Contents


**1 About ActivVisionTools** **1**

1.1 Introducing ActivVisionTools . . . . . 2

1.2 Basic Concepts . . . . . 4

1.3 The Family of Tools . . . . . 10

1.4 System Requirements . . . . . 14

**2 Installation** **15**

2.1 Installing ActivVisionTools . . . . . 16

2.2 Licensing . . . . . 20

**3 Using ActivVisionTools** **25**

3.1 Configuring Visual Basic 6.0 and Visual Basic .NET . . . . . 26

3.2 At Design Time . . . . . 32

3.3 Displaying Images . . . . . 34

3.4 Connecting to an Image Acquisition Device . . . . . 36

3.5 Creating a Stand-Alone Application . . . . . 38

3.6 Checking the Necessary Licenses . . . . . 40

3.7 Deploying an Application . . . . . 42

**4 Tips & Tricks** **43**

4.1 Error Messages and the Status of an Application . . . . . 44

4.2 Installing the Dongle Driver Manually . . . . . 44

4.3 Troubleshooting . . . . . 46

# Chapter 1

# About ActivVisionTools

This chapter will introduce you to the features and the basic concepts of ActivVisionTools. Furthermore, it introduces the members of this family of tools.


**1.1 Introducing ActivVisionTools . . . . . 2**
**1.2 Basic Concepts . . . . . 4**
1.2.1 The Different Types of Tools . . . . . 4
1.2.2 Interaction Via Graphical User Interfaces . . . . . 6
1.2.3 A Look Behind the Tools . . . . . 6
1.2.4 The Description File (DSC file) . . . . . 9
1.2.5 The Three Modes of Working with ActivVisionTools . . . . . 10
**1.3 The Family of Tools . . . . . 10**
**1.4 System Requirements . . . . . 14**

## 1.1 Introducing ActivVisionTools

Congratulations! By choosing ActivVisionTools, you got leading-edge vision algorithms inside an easy-to-use working environment – powerful machine vision applications can be developed quickly and easily, without writing a single line of code! Nevertheless, ActivVisionTools does not form a closed development environment but provides a documented programming interface that lets you integrate additional functionality very flexibly.

### ActivVisionTools and HALCON

ActivVisionTools is based upon HALCON, MVTec's unique machine vision library that offers highest performance together with unmatched flexibility. ActivVisionTools inherits the speed, robustness, and accuracy of HALCON, but hides the complexity of image processing from the user. Instead, the tools present solutions for standard tasks such as measuring or gauging objects, reading bar codes, or blob analysis.

If the available ActivVisionTools do not suffice to solve your machine vision task, you can combine them with HALCON, i.e., create your own ActivVisionTool based on HALCON code and use it together with the other ones. More information about this can be found in the Advanced User's Guide for ActivVisionTools, chapter 4 on page 107.

### ActivVisionTools and Programming Environments

ActivVisionTools comes as a set of *ActiveX controls*; thus, it can be used in all environments that can deal with ActiveX controls. As Visual Basic can be seen as the "native" environment for ActiveX, most examples in the ActivVisionTools manuals stem from this environment. They are based on version 6.0, as you can convert projects from this version to Visual Basic .NET but not the other way round. The basic steps are also shown in this manual for Visual Basic .NET.

Since Visual Basic 6.0 and Visual Basic .NET differ regarding the programming language, the basic examples using the ActivVisionTools programming interface are provided for both versions. Detailed information about using ActivVisionTools in Visual Studio .NET and Visual C++ is given in the Advanced User's Guide for ActivVisionTools in section 1.2 on page 3 and section 1.3 on page 14, respectively.

Please note that ActivVisionTools are only available as ActiveX controls, not as *.NET Windows Controls*! The support of .NET is therefore restricted to the use of ActiveX controls within a .NET environment.

In fact, ActivVisionTools is more than a set of ActiveX controls: It also includes further software components that automatically connect the tools to each other and configure the data streams between them.

### Outside ActivVisionTools

A real machine vision application has its input and output. The main input consists, of course, of the images to process. To facilitate the task of image acquisition as much as possible, the tool ActivView provides interfaces to about 50 image acquisition devices (see the manual of ActivView for a list). The connection to a frame grabber or camera is set up automatically; you simply specify the used camera/frame grabber configuration and the desired grabbing mode interactively and start to grab images.

Concerning the output, the ActivVisionTools family currently offers tools to output numerical results such as measurements together with evaluations like "is okay" to log files, via a serial interface, or via a digital I/O board (see the manuals of ActivSerial and ActivDigitalIO for more information). Output to other devices, e.g., your specific hardware, can be realized via the programming interface.

Using ActivDigitalIO, you can also synchronize your application with digital input signals.

### The Programming Interface of ActivVisionTools

Even if no programming is required to use ActivVisionTools, you can of course access the corresponding controls and classes from a programming environment like Visual Basic and integrate your own code. The programming interface is described in detail in the **Reference Manual**, which can be found in the Windows start menu (sub-folder `Start` ▷ `Programs` ▷ `MVTec ActivVisionTools 3.2` ▷ `Documentation`).

Examples for using the ActivVisionTools programming interface to access results and evaluations are described in the User's Manuals for ActivMeasure (see section 4.3 on page 30), ActivBarcode (see section 4.3 on page 26), ActivDataCode (see section 4.3 on page 40), ActivFeatureCalc (see section 4.2 on page 34), ActivOCR (see section 4.3 on page 62), ActivAlignment (see section 4.3 on page 28), and ActivDecision (see section 3.4 on page 30). The Advanced User's Guide for ActivVisionTools contains examples showing how to create a customized graphical user interface (see chapter 2 on page 39), how to influence the execution of ActivVisionTools (see chapter 3 on page 53), and how to create new ActivVisionTools based on ActivGenericInterface (see chapter 4 on page 107).

### Native Language Support

ActivVisionTools provides the so-called *Native Language Support*, i.e., the underlying software allows to switch the language of, e.g., the text labels in the ActiveX controls. Currently, ActivVisionTools supports the languages english, spanish, french, italian, german, japanese, korean, chinese, simplified chinese, and turkish. Please ask your local distributor about developments concerning other languages.

ActivVisionTools allows to *switch the language dynamically*. Therefore, you do not need to

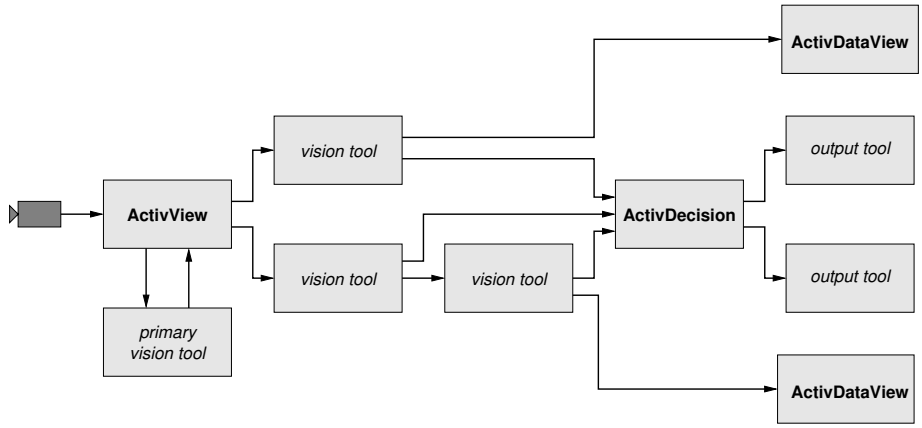


Figure 1.1: Example configuration of tools.

decide once and forever which language you want to use, instead you can switch between languages as you like. This also means that you can develop an ActivVisionTools application in one language and then switch the language for its final use.

## 1.2 Basic Concepts

This section explains the basic concepts behind ActivVisionTools: the different groups of tools and how they cooperate with each other, how to interact with a tool via its graphical user interfaces, how to put together suitable tools to create an application, and the different modes for designing and executing an ActivVisionTools application.

### 1.2.1 The Different Types of Tools

ActivVisionTools come in different "flavors": *basic tools*, *vision tools*, *primary vision tools*, *result processing tools*, *output tools*, and *inspection tools*.

*basic tools*

- *Basic tools* provide the functionality that is needed in each machine vision application. There is only one *basic tool*: ActivView. This tool is in charge of *displaying* the current image together with regions of interest (ROIs) and image processing results. Furthermore, ActivView possesses *interfaces to about 50 image acquisition devices* so that you can connect to your device and grab images in less than five mouse clicks. Finally, ActivView allows to specify *regions of interest* (ROIs) for other ActivVisionTools in order to focus and speed up image processing.

- *Vision tools* provide the functionality specifically needed for the machine vision application at hand, i.e., bar code reading, measuring (gauging), or blob analysis. In most applications, only one *vision tool* is needed. *vision tools*

  As input, *vision tools* typically receive the current image respectively regions of interest in it from ActivView, together with calibration and alignment information if present (see below). But *vision tools* can also be connected to each other in a chain, e.g., ActivFeatureCalc expects extracted "blobs" as an input, which stem from ActivBlobFinder.

  *Vision tools* can provide two types of results: *iconic* and *alphanumeric or symbolic* results. Iconic results are, e.g., preprocessed images or regions ("blobs"). Typical alphanumeric results (also called features) are distances measured in the image (or in 3D), a recognized character, or a decoded bar code; symbolic results could be error messages like `'no bar code found'` or `'part is not okay'`.

- *Primary vision tools* are not directly responsible for solving your vision task, but help you to setup the complete application. *primary vision tools*

  *Primary vision tools* fall into two categories: *calibration tools* and *change detectors*. *Calibration tools* estimate (calibrate) parameters of the vision system, e.g., to compensate for non-perfect optics or to transform measurements from pixels to 3D coordinates. All they need is a set of images of a special calibration pattern, which can be taken during the initial setup. The extracted information is sent back to ActivView, which in its turn adapts the live images accordingly or passes on the 3D information to the vision tools. Currently, ActivVisionTools provides one *calibration tool*: ActivGeoCalib.

  *Change detectors* allow to react during run time to changes in the environment, e.g., align a line of measurement to a changing position of the part to measure or adapt the image to a changing illumination. *Change detectors* have two modes of operations: During setup, they are trained on example images; during run time, they detect changes between the training images and the actual ones. This information is sent to ActivView, which in its turn adapts the current image or modifies the position of regions of interest. Currently, ActivVisionTools provides one *change detector*: ActivAlignment.

- *Result processing tools* allow to further process alphanumeric results. By default, *result processing tools* have access to all results of all *vision tools*. *result processing tools*

  ActivVisionTools includes one *result processing tool*: ActivDecision. This tool allows to evaluate the results of other tools by formulating conditions like "the measured distance must lie between certain values in order for the part being okay". These evaluations are then treated as additional results, therefore you can output them together with the alphanumeric results. Moreover, depending on these evaluations you can decide what is to be written to a log file or sent via the serial interface.

- *Output tools* allow to open and initialize channels to output alphanumeric or symbolic results. Depending on evaluations, you can select the output data. Currently, tools are provided for output into file, via a serial interface, and via a digital I/O board. *output tools*

A special kind of *output tools* are ActivDataView and ActivFeatureHistogram: Those two tools do not perform "real" output but display the alphanumeric results in a spreadsheet or a histogram, respectively. They are very useful when setting up and configuring an ActivVisionTools application, as you can check at once the validity of the results.

*inspection tools*

- *Inspection tools* allow to check the setup of camera, optics, or illumination. They can be used for the initial configuration of a vision system, but also for monitoring it during run time.

  Currently, ActivVisionTools includes three *inspection tools*, which allow to zoom into the current image, to inspect the gray value profile along a line or arc, and to calculate gray value histograms.

The example configuration of tools in figure 1.1 on page 4 illustrates the typical connections between ActivVisionTools.

Note that the following tools defy the above classification: ActivGenericInterface and ActivGUIComponents. ActivGenericInterface can be used to create your own ActivVisionTool; therefore, you can configure it to play the role of any of the tool types described above. ActivGUIComponents, on the other hand, is a set of components that allow to create customized GUIs.

### 1.2.2 Interaction Via Graphical User Interfaces

ActivVisionTools can be configured and parameterized completely via graphical user interfaces, without having to write a single line of code. To facilitate this interaction even more, each ActivVisionTool provides multiple ActiveX controls which "concentrate" on different groups of properties. In other words, each ActivVisionTool consists of multiple sub-tools: The so-called *master tool* contains the full functionality together with a window to configure the main parameters, while the so-called *support tools* provide more detailed access to the parameters of the *master tool*. Thus, you can adapt your working environment easily to the different phases of development. As an example, figure 1.2 on page 8 displays sub-tools of ActivView.

*Support tools* can be either placed on the form at design time, or opened at run time via a click with the right mouse button on the corresponding *master tool*. How you can further adapt the GUI to your needs is described in the Advanced User's Guide for ActivVisionTools, chapter 2 on page 39.

### 1.2.3 A Look Behind the Tools

Learning about the different tools, sub-tools, and their connection, you might start to wonder whether you really need no programming to get all these tools working together.

The reason why you really do not need to write a single line of code to work with ActivVision-Tools is that it already comes with built-in mechanisms that serve as a sort of operating system. In fact, you do not need to know much about these mechanisms, only that they

- automatically connect the tools and their data streams to each other,
- automatically connect *support tools* to their *master tool*,
- set up a correct order of execution for the tools (scheduling),
- monitor the application for events, e.g., user interactions, and trigger suitable reactions,
- save and restore the state of your ActivVisionTools application (see also the next section).

Thus, you can concentrate on your machine vision problem at hand.

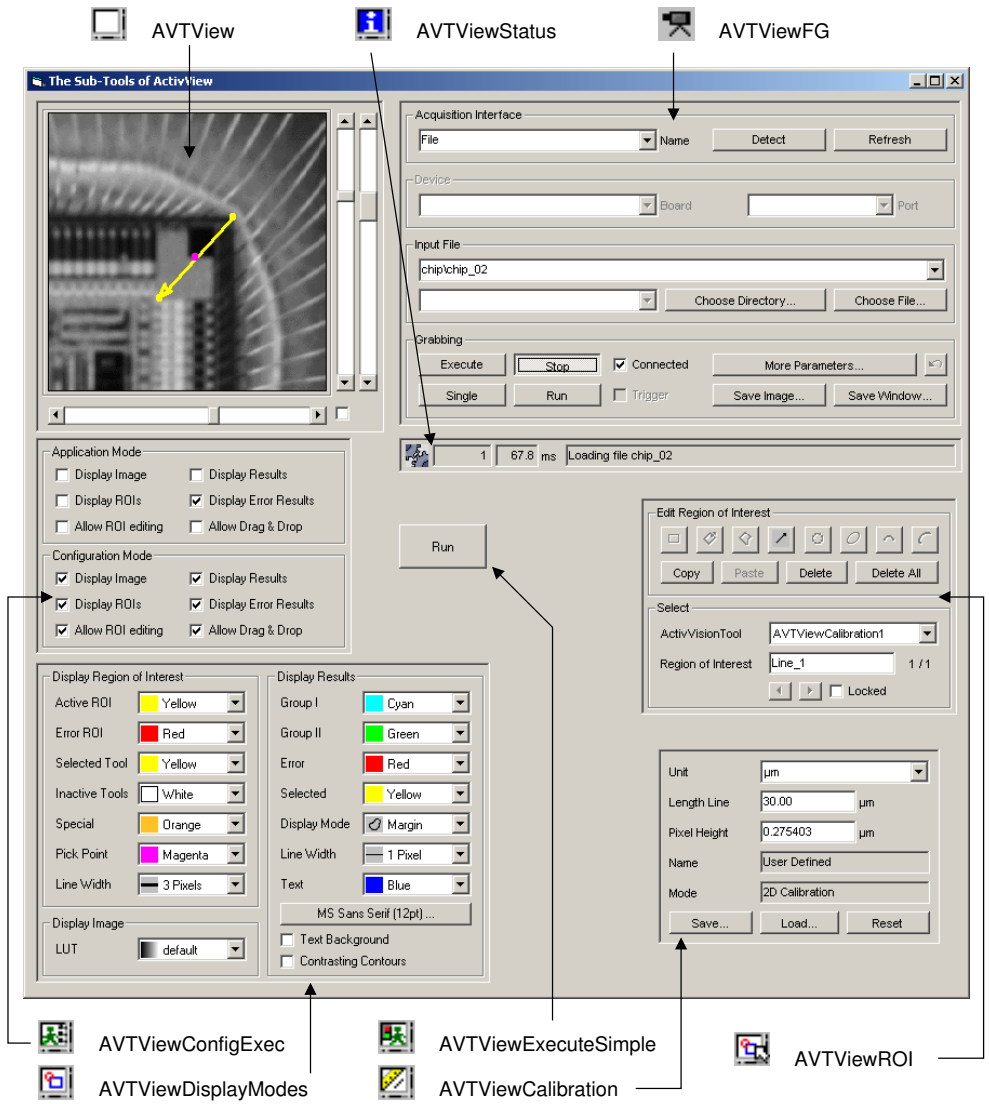


Figure 1.2: The sub-tools of ActivView .

### 1.2.4 The Description File (DSC file)

ActivVisionTools automatically saves the *state* of a project or executable, i.e., the selected values of the properties but also the number and position of the regions of interest, in the so-called *description file*. This is done not only when you save the project, but also whenever you start or stop an ActivVisionTools application and also when you exit from the developing environment. Please note that this is not completely true when using ActivVisionTools within Visual C++ or Visual Studio .NET. More details can be found in the Advanced User's Guide for ActivVisionTools, section 1.1 on page 2.

Since the DSC files are saved automatically, it may be useful to create a backup version of it before experimenting with the configuration of your application. Then, you can easily revert to the previous state without resetting the parameters manually.

Description files are marked by the file extension `.dsc`. By default, they get the name of the corresponding executable and are placed in the so-called *start directory* of the executable. For a "normal" executable the start directory corresponds to the directory where the executable actually resides; if you start an executable via a Windows shortcut, the start directory of the shortcut can be configured in the dialog which pops up when you click on the shortcut with the right mouse button and select `Properties`.

This simple strategy has two implications: First, if you change the start directory (by moving the executable or by configuring it directly for a shortcut), you must also move the description file accordingly, otherwise the application will not find it and start with default values (and without ROIs).

The second implication is not as obvious as the first: While developing an ActivVisionTools application within Visual Basic (version 6.0), the developing environment itself is the executable; thus, the description files would be saved under the name `VB6.dsc`. Of course, this is not the behavior you would expect. Therefore, the ActivVisionTools distribution includes an *Add-In* for Visual Basic called `AVTProjectManager`. This Add-In determines the name and the start directory of a Visual Basic project. This information is then used by ActivVisionTools when creating a description file for a Visual Basic project. Until a project is saved for the first time, `AVTProjectManager` puts the DSC file into the directory specified in the environment variable `TEMP`.

If an executable is created from a Visual Basic project (see section 3.5 on page 38) such that it is stored in another than the project directory, this new directory will be used as the new start directory. This may lead to the loss of some settings. To further use the current settings of the project, the DSC file must be copied manually to the new directory. Therefore, we recommend to create executables only in the project directory.

The programming interface of ActivVisionTools provides you with additional functionality for handling DSC files. For example, you can switch off the automatic saving of the state of your application or save it at arbitrary moments. Furthermore, you change the state by reading a

specified DSC file. For more information, please refer to the Advanced User's Guide for ActivVisionTools, section 3.5 on page 83.

### 1.2.5 The Three Modes of Working with ActivVisionTools

The concept of ActiveX knows two modes: At design time, you can place ActiveX controls onto a form, edit their shape and appearance, and change properties in the *Property Window*. By executing an application, you automatically switch to run time. In this mode, you interact with ActiveX controls via their graphical user interfaces.

In ActivVisionTools, this concept has been extended by offering two execution modes: *configuration mode* and *application mode*. This allows you to create two versions of your application: one for its initial configuration and one for the final, running system.

## 1.3 The Family of Tools

Currently, the following ActivVisionTools are available:

- **ActivView**
  Interfaces to about 50 image acquisition devices, regions of interest (ROIs), display of images, ROIs, and results, 2D calibration, execution control.

  How to use ActivView is described in the User's Manual for ActivView. Since ActivView is part of every ActivVisionTools application, its use is also described in other User's Manuals.

- **ActivZoom**
  Zooming of live images, inspection of pixel values.

  Currently, there is no separate manual for ActivZoom. The User's Manual for ActivBlobFinder, chapter 2 on page 9, describes how to use ActivZoom to determine suitable parameters for the extraction and the processing of blobs.

- **ActivLineProfile**
  Analysis of pixel values along lines or arcs, calculation of statistical information, e.g., the mean pixel value.

  Currently, there is no separate manual for ActivLineProfile. The User's Manual for ActivView, section 4.1 on page 40, describes how to use ActivLineProfile to analyze and enhance the quality of acquired images. The User's Manual for ActivBlobFinder, chapter 2 on page 9, describes how to use ActivLineProfile to determine suitable parameters for the extraction and the processing of blobs.

- **ActivHistogram**
  Analysis of the distribution of gray values.

  Currently, there is no separate manual for ActivHistogram. The User's Manual for ActivView, section 4.1 on page 40, describes how to use ActivLineProfile to analyze and enhance the quality of acquired images. Further examples can be found in the Advanced User's Guide for ActivVisionTools, section 3.4.2.2 on page 80 and section 4.4.1 on page 119.

- **ActivGeoCalib**
  Geometric camera calibration for 3D measurements.

  How to use ActivGeoCalib is described in the User's Manual for ActivGeoCalib. Further examples for using ActivGeoCalib can be found in the Advanced User's Guide for ActivVisionTools, section 3.6.2.3 on page 105 and section 4.4.3.1 on page 137.

- **ActivAlignment**
  Automatic alignment of ROIs to moving parts.

  How to use ActivAlignment is described in the User's Manual for ActivAlignment. Further examples for using ActivAlignment can be found in the User's Manual for ActivDecision and in the Advanced User's Guide for ActivVisionTools, section 3.6.2.2 on page 104 and section 4.4.3.2 on page 141.

- **ActivMeasure**
  Measurements along lines or arcs with subpixel accuracy.

  How to use ActivMeasure is described in the User's Manual for ActivMeasure. Further examples for using ActivMeasure can be found in the User's Manual for ActivAlignment, e.g. in chapter 3 on page 15, in the User's Manual for ActivGeoCalib, chapter 3 on page 17, in the User's Manual for ActivDecision, and in the Advanced User's Guide for ActivVisionTools, section 3.4.1.1 on page 68, section 3.4.2.2 on page 80, section 3.6.2.3 on page 105, section 4.4.1 on page 119, and section 4.4.3 on page 137.

- **ActivBarcode**
  Extraction and decoding of various common bar codes.

  How to use ActivBarcode is described in the User's Manual for ActivBarcode. Further examples for using ActivBarcode can be found in the User's Manual for ActivView, chapter 3 on page 27, in the User's Manual for ActivSerial, and in the Advanced User's Guide for ActivVisionTools, section 1.3 on page 14, section 2.2 on page 42, section 3.6.1.4 on page 98, and section 3.4.1.1 on page 68.

- **ActivDataCode**
  Extraction and decoding of various common data codes.

  How to use ActivDataCode is described in the User's Manual for ActivDataCode.

- **ActivBlobFinder**
  Robust blob extraction and further processing of blobs.

How to use ActivBlobFinder is described in the User's Manual for ActivBlobFinder. Further examples for using  can be found in the User's Manual for ActivFeatureCalc and in the Advanced User's Guide for ActivVisionTools, section 3.4.1.2 on page 72, section 3.4.2.1 on page 76, section 3.4.2.2 on page 80, section 3.6.1.2 on page 94, section 3.6.1.3 on page 96, and section 4.4.1 on page 119.

- **ActivFeatureCalc**
  Calculation of more than 50 shape and gray value features for blob analysis.

  How to use ActivFeatureCalc is described in the User's Manual for ActivFeatureCalc. Further examples for using ActivFeatureCalc can be found in the User's Manual for ActivBlobFinder.

- **ActivOCR**
  Easy-to-use, fast and reliable OCR.

  How to use ActivOCR is described in the User's Manual for ActivOCR . Further examples for using ActivOCR can be found in the Advanced User's Guide for ActivVisionTools, section 3.6.1.1 on page 92, section 3.6.1.4 on page 98, and section 3.6.2.2 on page 104.

- **ActivWordProcess**
  Grouping characters to words and checking them against syntax rules or dictionaries .

  How to use ActivWordProcess is described in the User's Manual for ActivOCR, chapter 3 on page 41. Further examples for using ActivWordProcess can be found in the Advanced User's Guide for ActivVisionTools, section 3.6.1.1 on page 92, section 3.6.1.4 on page 98, and section 3.6.2.2 on page 104.

- **ActivDataView**
  Display of alphanumeric results.

  Currently, there is no separate manual for ActivDataView. Instead, the following User's Manuals describe how to use ActivDataView to view the results of other ActivVisionTools: ActivGeoCalib (see section 3.1 on page 18), ActivAlignment (see section 3.1 on page 16), ActivMeasure (see e.g. section 2.3 on page 14), ActivBarcode (see e.g. section 2.2 on page 8), ActivBlobFinder (see section 3.1 on page 26), ActivFeatureCalc (see e.g. section 2.2 on page 20), and ActivOCR (see section 2.4 on page 38).

- **ActivFeatureHistogram**
  Histogram of features, i.e., alphanumeric results.

  Currently, there is no manual for ActivFeatureHistogram.

- **ActivDecision**
  Interactive formulation of conditions to evaluate results and display of individual results and evaluations.

  How to use ActivDecision is described in the User's Manual for ActivDecision. Furthermore, the following User's Manuals describe how to use ActivDecision to evaluate the

results of other ActivVisionTools: ActivAlignment (see section 3.2 on page 18), ActivMeasure (see section 3.2 on page 20), ActivBarcode (see section 3.2 on page 16), ActivBlobFinder (see section 3.2 on page 28), ActivFeatureCalc (see section 3.2 on page 26), and ActivOCR (see section 3.2 on page 52).

How to access evaluation results via the programming interface is described in the following User's Manuals: ActivMeasure (see section 4.3.2 on page 36), ActivBarcode (see section 4.3.2 on page 32), ActivFeatureCalc (see section 4.2.2 on page 41), and ActivOCR (see section 4.3.2 on page 67).

- **ActivFile**
  Selection of results for output and customization of the log file format.

  Currently, there is no separate manual for ActivFile. Instead, the following User's Manuals describe how to use ActivFile to output the results of other ActivVisionTools: ActivAlignment (see section 3.3 on page 20), ActivMeasure (see section 3.3 on page 22), ActivBarcode (see section 3.3 on page 18), ActivBlobFinder (see section 3.3 on page 30), ActivFeatureCalc (see section 3.3 on page 28), and ActivOCR (see section 3.3 on page 54).

- **ActivSerial**
  Selection of results for output and configuration of the communication via serial interface (RS232).

  How to use ActivSerial is described in the User's Manual for ActivSerial.

- **ActivDigitalIO**
  Input and output via a digital I/O board.

  How to use ActivDigitalIO is described in the User's Manual for ActivDigitalIO.

- **ActivGenericInterface**
  Tool without own activity, which allows to integrate your own functionality.

  How to use ActivGenericInterface is described in the Advanced User's Guide for ActivVisionTools, chapter 4 on page 107.

- **ActivGUIComponents**
  The ActivVisionTools version of scroll bars, combo boxes etc. that allow to create customized GUIs interactively.

  How to use ActivGUIComponents is described in the Advanced User's Guide for ActivVisionTools, chapter 2 on page 39.

- **ActivHALCONFoundation**
  A virtual tool providing a license to use HALCON within ActivVisionTools applications. Note that ActivHALCONFoundation does not comprise the full functionality of HALCON; please refer to the Advanced User's Guide for ActivVisionTools, section 4.2 on page 108, for more information.

## 1.4 System Requirements

To use ActivVisionTools, you need a platform with at least:

- Microsoft Windows NT 4.0, Windows 2000 or Windows XP, or Windows Vista
- Microsoft Visual Basic 6.0
  Microsoft Visual C++ 6.0
  Microsoft Visual Studio .NET 2002 or 2003
  Microsoft Visual Studio 2005 or 2008 (including Express versions)
- Intel Pentium processor (or compatible)
- 150 MB free disk space (run-time version; full version: 200 MB)
- 256 MB memory (Visual Studio .NET: 512 MB recommended)
- 16 bit display with 1024×768 pixel

! Please note that **each developer using ActivVisionTools needs at least `'Power User'` privileges**! To use ActivVisionTools within Visual Studio .NET, you must also be a member of the group `'Debugger User'`.

If you want to develop .NET projects in other locations than on a local disk, you must configure the .NET security (via the command line tool `caspol` or the "Microsoft .NET Framework Configuration", which can be opened via the Administrative Tools in the Control Panel) to allow executing code from your desired project location. Otherwise, you get a warning upon creating or loading the project that the location is not trusted and a run-time error upon executing it.

# Chapter 2

# Installation

This chapter will show you how to install ActivVisionTools on your system and explain the different licensing schemes. Thus, at the end of this chapter you will be all set for creating your first ActivVisionTools application.


**2.1 Installing ActivVisionTools . . . . . 16**
2.1.1 The Installation Process . . . . . 16
2.1.2 The Resulting File Structure of ActivVisionTools . . . . . 18
2.1.3 Additional Information for HALCON Users . . . . . 19
2.1.4 Uninstalling ActivVisionTools . . . . . 19
**2.2 Licensing . . . . . 20**
2.2.1 The License for ActivView and the inspection tools . . . . . 20
2.2.2 The Evaluation License . . . . . 20
2.2.3 Permanent Licenses . . . . . 20

## 2.1 Installing ActivVisionTools

Before describing how to install ActivVisionTools, let's take a look at the different licensing schemes and the relation between ActivVisionTools and HALCON.

### Some Words about Licensing

Independent of whether you install ActivVisionTools from the DVD or from the Web, the installed version includes an unlimited license key which lets you use ActivView and the *inspection tools* (ActivZoom, ActivLineProfile, ActivHistogram) without further ado. This means you can grab images from your image acquisition device and inspect them free of charge! The license key is automatically placed in the file `license.dat` which will be installed in the subdirectory `license` of the folder you choose for ActivVisionTools during the installation.

To evaluate the full functionality of ActivVisionTools you can obtain *temporary license keys* from you local distributor free of charge. You will receive the keys in form of a new license file `license.dat` to replace the original one. An evaluation license is valid for one month.

After the evaluation phase, you will have to purchase *permanent license keys* for those tools (beside ActivView, ActivZoom, ActivLineProfile, and ActivHistogram) you want to use. Again, these keys will come in form of a new license file `license.dat`. Licenses are bound to one specific host computer or one hardware dongle.

Licensing is described in more detail in section 2.2 on page 20.

### ActivVisionTools and HALCON

ActivVisionTools is based on HALCON, MVTec's unique machine vision library. Therefore, the installation process will automatically install the required components of HALCON if necessary, i.e., if it does not find a current HALCON version on your system. Note that you do not need a HALCON license to run ActivVisionTools.

### 2.1.1 The Installation Process

! Please note that **you need administrator privileges in order to install ActivVisionTools!**

To install ActivVisionTools, simply insert the DVD. This should automatically start the setup program. If the setup program doesn't start automatically (e.g., because you switched off this feature on your system), execute the program `Setup.exe` located in the directory `install-windows` on the DVD.

At the beginning, you will be asked to select a folder where the ActivVisionTools software will be installed. A typical place would be `C:\Program Files\MVTec\ActivVisionTools` . In

the following, we will call this folder the *base directory*. Furthermore, the setup will ask for the name of a program folder in the Windows start menu where it will add program icons, e.g., the user's manuals or the ready-to-use example application ActivViewer. By default, these icons will appear in the submenu `Start` ▷ `Programs` ▷ `MVTec ActivVisionTools 3.2`.

The setup allows you to choose between different types of installations:

1. **Typical:** All parts of ActivVisionTools (recommended for most users).
2. **Custom:** Select the desired parts.
3. **Runtime:** Minimal installation (requires less disk space).

Then, the setup starts to install the required components of HALCON. If you are already using HALCON, please make a quick detour to section 2.1.3 on page 19 at this point. Again you will be asked to select a folder, this time for the HALCON software; by default, it will be installed "parallel" to the ActivVisionTools software.

Then, the setup asks whether to install the dongle driver programs which are necessary for using a license bound to a dongle. We recommend to install the dongle drivers. Note that **this choice only concerns the permanent licenses!** If you just want to use ActivView, ActivZoom etc. or evaluate ActivVisionTools with a temporary license, we nevertheless recommend to install the dongle driver to facilitate a later use of permanent licenses.

If you choose to install the dongle drivers, the corresponding setup will be launched automatically, without any further action necessary from your part. If you do not install the dongle drivers when installing ActivVisionTools, you may install them manually afterwards; for details please refer to section 4.2 on page 44.

Now, the HALCON and ActivVisionTools run-time environments are actually installed on your system. On some systems a reboot may be necessary after the installation.

### 2.1.2 The Resulting File Structure of ActivVisionTools

After the installation, the Windows start menu contains the folder `Start` ▷ `Programs` ▷ `MVTec ActivVisionTools 3.2`, which lets you open the user's manuals or uninstall ActivVisionTools (see also section 2.1.4). Furthermore, you can start `ActivViewer`, which is a ready-to-run application consisting of ActivView, ActivZoom, ActivLineProfile, and ActivHistogram, and some ready-to-run eample applications.

Let's have a quick look at the file structure which results after the installation in the base directory of ActivVisionTools:

- `bin`: The subdirectory `x86-win32` of this directory contains the ActivVisionTools OCX `ActivVTools.ocx` plus additional libraries. Furthermore, it contains the precompiled ActivView application `ActivViewer.exe` (if installed).
- `devenv`: This directory contains special extension you need when using ActivVisionTools in Visual C++ or Visual Studio .NET. For details see the Advanced User's Guide for ActivVisionTools, chapter 1 on page 1.
- `doc`: If you installed the documentation, you can find the User's Manuals in form of PDF files in the subdirectory `doc\pdf\manuals`. The reference manual is available as compiled help and can be found in the subdirectory `doc\chm`.
- `examples`: The subdirectory `demo` contains ready-to-use example applications. In the subdirectories of `examples\manuals` you can find the example Visual Basic 6.0 projects that allow you to follow the examples in the User's Manuals actively. Examples for using ActivVisionTools in Visual C++ and Visual Studio .NET can be found n the subdirectories of `examples\vcpp` and `examples\dotnet`, respectively. We recommend to **create a private copy of the example projects before working on them!**

  !
- `images`: This directory contains image files (TIFF format) which are used in the examples in the user's manuals.
- `license`: The license file `license.dat` resides in this directory. After the installation this file contains the unlimited license key for ActivView, ActivZoom, ActivLineProfile, and ActivHistogram (see section 2.2 on page 20 for more information).
- `nls`: As mentioned in chapter 1 on page 1, ActivVisionTools supports different languages (Native Language Support). For this, the installation process checks the language used by the operating system on your computer and installs the corresponding files containing, e.g., the text for the labels, in the directory `nls`. The files correspond to the ActiveX controls; their name consists of the name of the control and the extension `.nls`, e.g., `AVTView.nls`.

  If you use a language which is not supported yet, the english version will be installed. You can switch the language by copying the corresponding language files from the DVD into this directory.

- `ocr`: This directory contains special resources for ActivOCR (see the User's Manual for ActivOCR ).
- `res`: This directory contains internally used AVI files.

### 2.1.3 Additional Information for HALCON Users

ActivVisionTools 3.2 is based on HALCON 8.0.2. If you already installed HALCON 8.0 or 8.0.1, the ActivVisionTools setup will update this HALCON installation by replacing the modified parts. Otherwise it will install a runtime version of HALCON 8.0.

If the ActivVisionTools setup detects an older HALCON version, e.g., HALCON 7.1.x, it will ask you whether to upgrade, remove, or keep this version. In fact, you get the same behavior as if you installed HALCON 8.0.2 over an existing older version.

If you choose to keep an older version, you can switch between the versions by modifying the environment variable `%HALCONROOT%`: To use ActivVisionTools, you must set this variable to the base directory you selected for HALCON during the installation of ActivVisionTools. To use the older HALCON version, set the variable to the corresponding base directory.

Please note that even if you keep an older HALCON version, **you cannot use the older COM interface anymore!** The reason is that ActivVisionTools is based on the COM interface and therefore create corresponding entries in the Windows registry which are difficult to switch.

### 2.1.4 Uninstalling ActivVisionTools

You can uninstall ActivVisionTools by selecting the entry `Uninstall ActivVisionTools` in the Windows menu `Start` ▷ `Programs` ▷ `MVTec ActivVisionTools 3.2`. Please note that **you need administrator privileges in order to uninstall ActivVisionTools successfully!** If you try to uninstall ActivVisionTools without administrator privileges, the setup program might falsely state that the uninstallation was successful but leave some parts behind, e.g., in the Windows start menu.

## 2.2 Licensing

ActivVisionTools checks whether the product is licensed to the user with the help of the FLEXlm license manager package from Macrovision. This section describes how to obtain and install a license.

### 2.2.1 The License for ActivView, ActivZoom, ActivLineProfile, and ActivHistogram

In fact, by installing ActivVisionTools you already obtained your first license key: The free and unlimited license to use ActivView, ActivZoom, ActivLineProfile, ActivHistogram, ActivDataView, AFeatHisto, and ActivGenericInterface. The license file `license.dat` which contains this key resides in the subdirectory `license` of the ActivVisionTools base directory (see section 2.1.2 on page 18) and is depicted in figure 2.1.

```
FEATURE MVTec_ActivView mvtecd 3.2 permanent uncounted HOSTID=ANY \
        SIGN="C45D 133D 66E3 BF25 3AFD 1860 FB2E 7C50 A0EE F2EE \
        A324 E9EF FF57 9DD0 4EA7 094A 1D3D 19C2 1678 F817 327E 1DC2 004C \
        8825 2D5E C7A8 A397 5F91 7A5E 98E2"
```

Figure 2.1: The permanent license key for ActivView, ActivZoom, ActivLineProfile, ActivHistogram, ActivDataView, AFeatHisto, and ActivGenericInterface .

This license also allows you to use the ready-to-run ActivVisionTools application ActivViewer.

### 2.2.2 The Evaluation License

To evaluate ActivVisionTools, you can obtain temporary license keys from your local distributor free of charge. You will be sent a new license file `license.dat` containing keys for the different ActivVisionTools as depicted in figure 2.2. Place this file in the subdirectory `license` of the ActivVisionTools base directory **on every computer you want to evaluate ActivVisionTools on** and rename it to `license.dat`, if necessary.

### 2.2.3 Permanent Licenses

If after the evaluation phase you decide to actually employ ActivVisionTools, you will have to purchase *permanent license keys* for those tools (beside ActivView, ActivZoom, ActivLineProfile, ActivHistogram, ActivDataView, ActivFeatureHistogram, and ActivGenericInterface) you want to use.

```
FEATURE MVTec_ActivSerial mvtecd 3.2 01-jun-2008 uncounted HOSTID=DEMO \
        SIGN="12BC 3012 369C 8E93 AB14 0BD3 5366 6CC6 BF4D DA53 2259 DAE8 \
        4139 2828 0D40 0C24 A33F 66C5 B428 CEAB 34AE 08BB 2B4C 2263 AD4B \
        01C3 1195 EAAC 2C0C DFBF"
FEATURE MVTec_ActivFile mvtecd 3.2 01-jun-2008 uncounted HOSTID=DEMO \
        SIGN="0C90 E735 F76B 4D3F 8D11 B567 9FBB 1E5F AAFD 74B8 9C26 EBF4 \
        9603 D10F F856 13DF DCDC 3A4F 0A1A C25E 1E60 2D75 2466 5978 FB5B \
        8796 C35D 62DF 7E92 A556"
FEATURE MVTec_ActivDecision mvtecd 3.2 01-jun-2008 uncounted HOSTID=DEMO \
        SIGN="1F37 4C4C 8D2B B78D FDBC 1701 1508 0D10 06FC 5147 A169 92DA \
        7226 DF21 895D 04A0 C4DF AD95 EA4B 82B7 1CD4 E95C B9AF 9455 9E99 \
        7AC0 1A99 6C07 5821 C361"
FEATURE MVTec_ActivBlob mvtecd 3.2 01-jun-2008 uncounted HOSTID=DEMO \
        SIGN="1663 AC1D A295 DDF9 0213 BEEC 4851 F2AB B506 90FA D22B DDB4 \
        EBC2 F393 CF61 1689 A8DA BDA9 62B9 B767 F7BB 5469 4DC1 AD66 1D81 \
        894D 3DCD EE5F C677 3B0E"
FEATURE MVTec_ActivBarCode1D mvtecd 3.2 01-jun-2008 uncounted HOSTID=DEMO \
        SIGN="1445 1DFE D4BE 4A13 A8A0 EB99 FD21 2605 7049 AFCD 2326 3739 \
        2CD4 63CB AB55 1236 406E 7B8F AB1A 1CF1 7232 F47D D6B7 D695 78C5 \
        2D54 1A7E D240 EA85 0B62"
FEATURE MVTec_ActivMeasure mvtecd 3.2 01-jun-2008 uncounted HOSTID=DEMO \
        SIGN="0385 85AC 46A9 5351 A1A6 EDBA 5398 92AD E2C5 7F6E 0F3F FE63 \
        01DF 8340 0A55 0FCB D0C9 0639 3AAE 6578 F27F 1B82 CE76 D754 1A93 \
        E457 BC88 014C 89CD 6929"
FEATURE ...
```

Figure 2.2: An example license file containing temporary license keys (not complete).

The first step to obtain a permanent license for specific ActivVisionTools is to select the computer on which you want to use ActivVisionTools on. Alternatively, you can decide to use a *hardware dongle*.

### Licenses Bound to a Computer

To obtain a license bound to a certain computer you must *extract identifying information* about this computer. The standard method for identifying an computer requires that the computer of your choice is equipped with a *network card*.

ActivVisionTools automatically extracts the IDs which are available on your system; they are displayed in ActivViewer in the menu item `About` ▷ `License Information`. For example, in figure 2.3, ActivVisionTools has detected a network card and depicts its host ID in the text box `Host ID`. Dongle IDs would show up in the text box `Flex ID`. If no strings appear although your computer does have a network board or a dongle, please refer to section 4.3.1 on page 46.

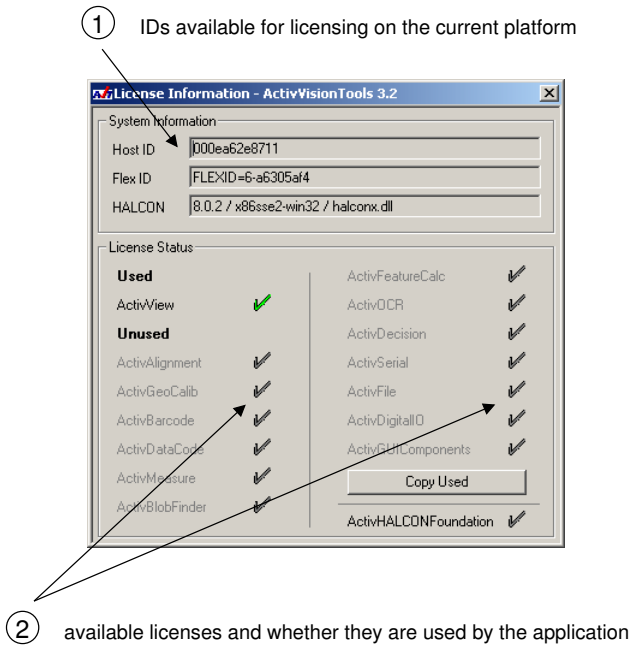


Figure 2.3: ActivViewer's dialog showing the available host IDs and licenses.

Send the ID to your local distributor together with the name of the host computer in your network and the names of the tools you want to use.

As you can see in the figure, the dialog also shows the available licenses (in the example: all tools) and which ones are actually used by the current application. The latter information is, of course, especially relevant for "real" ActivVisionTools applications. Therefore, you can open the dialog also via ActivView. This is shown in section 3.6 on page 40.

### Licenses Bound to a Dongle

You can also use a *hardware dongle* for licensing. In this case, the license will be coupled to this dongle, i.e., you can use ActivVisionTools on another computer by moving the dongle. This has the advantage that you don't commit yourself to using a certain computer. You can obtain such a dongle, for the parallel port or for the USB port, from your local distributor; you just need to specify which tools you want to use. The dongle will be delivered together with the corresponding license file. Note that dongles are more expensive than network cards.

```
FEATURE MVTec_ActivMeasure mvtecd 3.2 permanent uncounted \
        HOSTID=000ea62e7fa4 SIGN="8D65 03EB 10B7BF11 0C41\
        8528 150E 5D5F E1B5 1090 E96A 7CCB 7E6C 8946 2F39 1056 D711 DB3B \
        C239 C46F E2F5 245B 3DF0 B9F9 893B 5E8E C22C 1C65 3B99 AAE7"
FEATURE MVTec_ActivDecision mvtecd 3.2 permanent uncounted \
        HOSTID=000ea62e7fa4 SIGN="03EF A10E 6D69 67BB CAEA \
        3F50 3CE5 F48E 9E1F B1DB 5E6B EE25 FBC2 BCB4 3A32 14B1 B8D2 77B4 \
        A211 34ED 1BFB 34C7 BD33 54BE 69CF F9CF 577F 8AA0 A5AA 430C"
```

Figure 2.4: Permanent license keys, bound to a network board.

### Installing the License File

After sending the ID to your distributor (or requesting a dongle) you will receive a license file containing the permanent license keys for the requested tools. This file must be placed in the the subdirectory `license` of the ActivVisionTools base directory and renamed to `license.dat`, if necessary. Example license keys for ActivMeasure and ActivDecision bound to a network board with the ID `000ea62e7fa4` are depicted in figure 2.4.

Note that if you use a dongle-bound license and move the dongle to another computer, you must **copy the license file** to this computer as well!

If you upgrade ActivVisionTools to a newer version, your distributor provides you with a new license file, which contains the new license keys. This new license file should replace the old one.

# Chapter 3

# Using ActivVisionTools

In this chapter, we create your first example application. It addresses the elementary problem of all machine vision tasks: How to get images (from file or image acquisition device) and how to display and inspect them. For this task, only ActivView is needed. While solving this task, you will get to know the basic steps of working with ActivVisionTools: How to add (and remove) tools, how to adapt their properties, and how to test and execute the application.

The example application is developed within Microsoft Visual Basic. For each step, there is a Visual Basic 6.0 project in the subdirectory `examples\manuals\getting_started` of the ActivVisionTools base directory you selected during the installation (default: `C:\Program Files\MVTec\ActivVisionTools` ). However, we recommend to develop your own project from scratch, especially if you are using Visual Basic .NET.


**3.1 Configuring Visual Basic 6.0 and Visual Basic .NET . . . . . 26**
**3.2 At Design Time . . . . . 32**
**3.3 Displaying Images . . . . . 34**
**3.4 Connecting to an Image Acquisition Device . . . . . 36**
**3.5 Creating a Stand-Alone Application . . . . . 38**
**3.6 Checking the Necessary Licenses . . . . . 40**
**3.7 Deploying an Application . . . . . 42**

## 3.1 Configuring Visual Basic 6.0 and Visual Basic .NET

During the installation of ActivVisionTools, the corresponding OCX `ActivVTools.ocx` and other libraries are registered and therefore ready to use in a development environment. In this section, we show how to configure Microsoft Visual Basic 6.0 to use ActivVisionTools, followed by the procedure for Visual Basic .NET (Visual C# .NET is similar).

If you want to use ActivVisionTools within Visual C++, please refer to the Advanced User's Guide for ActivVisionTools, section 1.3 on page 14. More details about using ActivVisionTools in Visual Studio .NET can be found in the Advanced User's Guide for ActivVisionTools, section 1.2 on page 3.

### 3.1.1 Configuring Visual Basic 6.0

**Visual Basic Example**

Preparation for the following example:

- ☐ Start Microsoft Visual Basic and create a new project.

The following steps are visualized in figure 3.1.

1. For each new project, you must add ActivVisionTools to its components: Open the menu dialog `Project` ▷ `Components`, check the box labeled `ActivVisionTools 3.2`, and press `Apply`.
2. In the toolbar, new icons corresponding to the ActivVisionTools appear.
3. We recommend to move all ActivVisionTools icons into a separate tab in the toolbar. To create a tab, click into the toolbar with the right mouse button and select the corresponding entry. Then you can drag the icons into it with the left mouse button. This step needs to be performed only once.

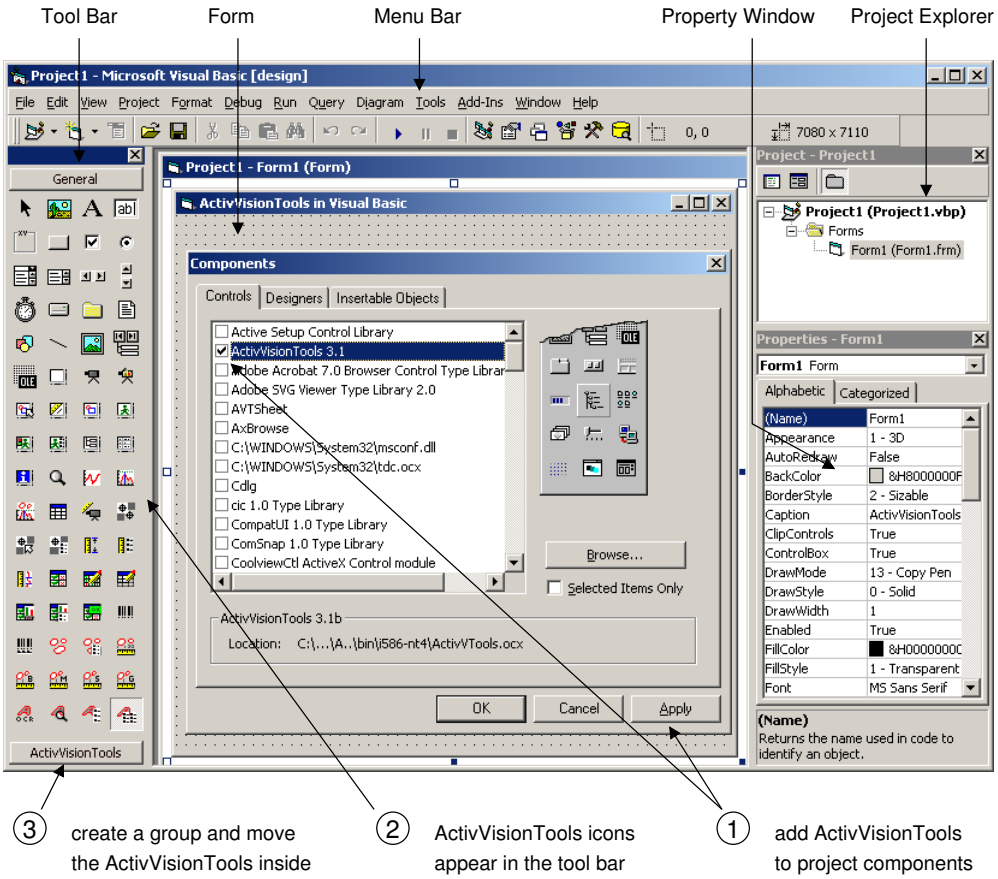


Figure 3.1: ActivVisionTools in the environment of Visual Basic 6.0.

### 3.1.2 Configuring Visual Basic .NET

This section describes the basic steps necessary to use ActivVisionTools in Visual Basic .NET. In contrast to Visual Basic 6.0, you need to perform them only once; then, you can use ActivVisionTools without further action in all new projects. If you get warnings or even run-time errors in the following, please check the system requirements listed in section 1.4 on page 14.

**Visual Basic .NET Example**

Preparation for the following example:

☐ Start Visual Basic .NET and create a new project of the type “Windows Application”.

The following steps are visualized in figure 3.2.

① To allow adding ActiveX controls to a Form using the Form Designer, the controls must first be added to the toolbox. This window is usually docked on the left side of Visual Studio .NET. If you don’t see it, press the keys `Ctrl`-`Alt`-`x` or select the menu item `View` ▷ `Toolbox`.

Right-click on the toolbox and select `Add Tab` in the context menu. This will create a new logical group for the ActivVisionTools ActiveX controls. Type “ActivVisionTools” as name for the new tab and confirm it by pressing `Enter`. Click on the new tab to open it. It should only contain the default item `Pointer`.

② To add the ActivVisionTools to the tab, right-click again on the toolbox and select `Customize Toolbox`. This will open a dialog displaying all available COM and .NET components. If necessary, select the tab `COM`. The list of displayed components is usually quite long. The ActivVisionTools are easy to locate because their names start with `ActivVisionTools.AVT` and appear close to the top of the list if the ordering is alphabetic, which is the default (otherwise, click on the table header `Name`).

To select the ActivVisionTools components, you must check all the boxes to the left of the corresponding names. The fastest way to do so is to select the first tool by clicking on `ActivVTools.AVTAlignment`. Then, scroll down to the last tool (`ActivVTools.AVTZoom`) and press the key `Shift` while clicking on it; all ActivVisionTools should be marked as selected. Now, click on the box to the left of `ActivVTools.AVTZoom`; as a result, all ActivVisionTools check boxes should become checked. Finally, click `OK` to add the tools to the toolbox.

③ As you can see, the list of tools is quite long. To get a more compact representation, right-click on the toolbox and disable the item `List View`.

We continue with the configuration on the next double page.

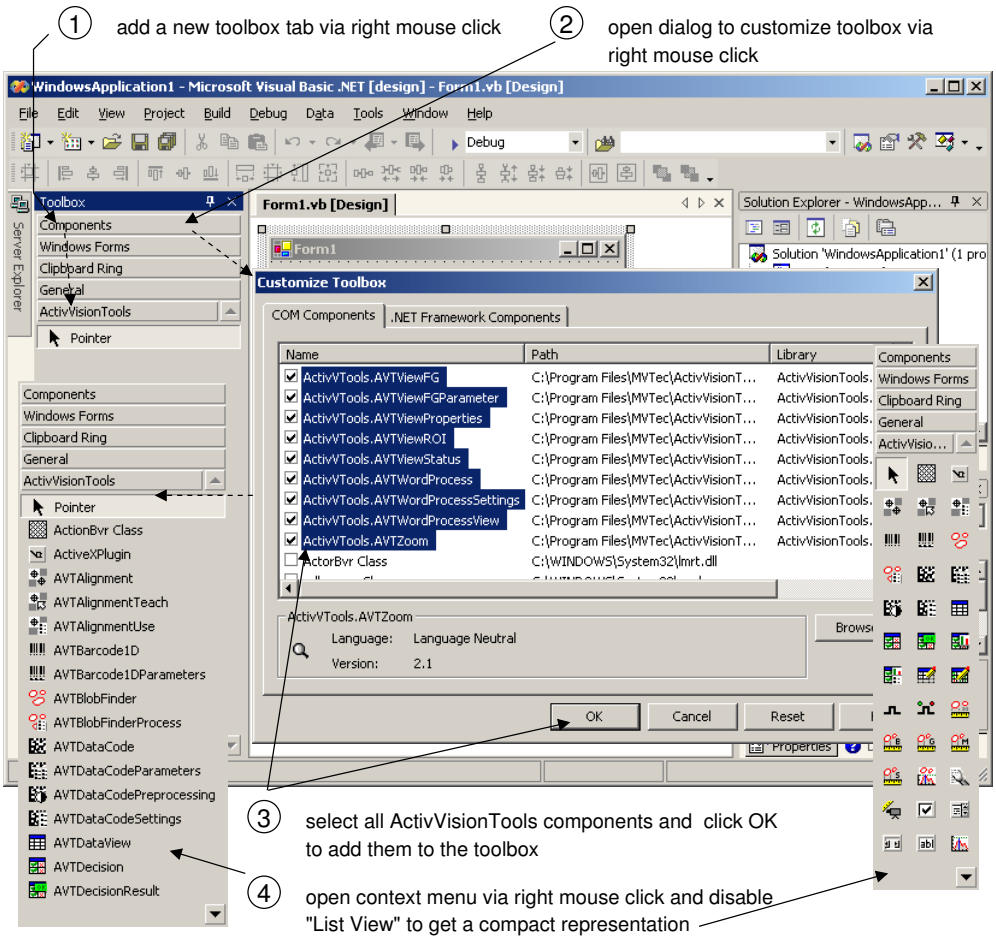


Figure 3.2: Adding ActivVisionTools to the Visual Studio .NET toolbox.

## (Configuring Visual Basic .NET, continued)

As described in detail in the Advanced User's Guide for ActivVisionTools in section 1.1 on page 2, Visual Basic .NET imposes some restrictions that made using ActivVisionTools not as convenient as in Visual Basic 6.0. In previous versions of ActivVisionTools, this meant that you had to add code to each ActivVisionTools application to perform garbage collection, save DSC files, and obtain modeless dialogs. Starting with ActivVisionTools 3.1, the ActivVisionTools fileset provides the necessary functionality in form of the control `AVTNETControl`, which can be added to the form interactively.

If for some reason you don't want to use the control but add (parts of) the functionality manually, please refer to Advanced User's Guide for ActivVisionTools, section 1.2.1 on page 3 and section 1.2.2 on page 4

The following steps are visualized in figure 3.3.

④ To add `AVTNETControl` to the toolbox, click on the tab containing the ActivVisionTools with the right mouse button and select `Customize Toolbox`.

⑤ In the dialog, select the tab `.NET Framework Components` and click on `Browse`. In the appearing file selection box, change to the subdirectory `devenv\dotnet\vb\ActivVToolsNET\bin` of the folder where you installed ActivVisionTools. Here, load `ActivVToolsNET.dll`. The previous dialog now lists the control as shown in the figure, and the toolbox contains its icon .

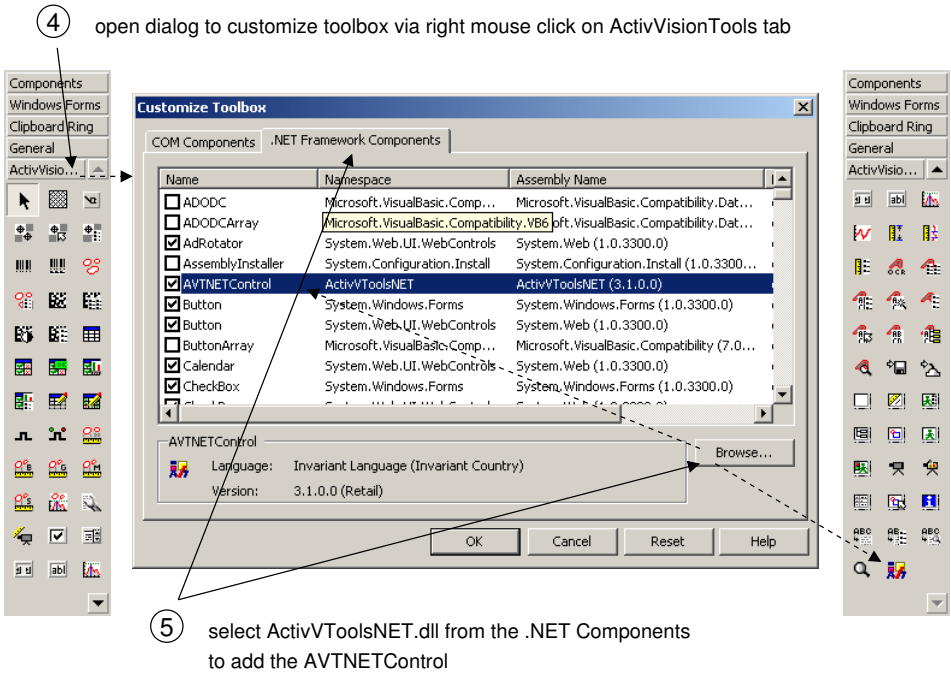


Figure 3.3: Adding ActivVisionTools to the Visual Studio .NET toolbox.

## 3.2 At Design Time

At design time, you create the layout of an application, i.e., you place the necessary elements on the form and edit their size and appearance. The first element of all ActivVisionTools applications is of course the window in which images and results are displayed. The corresponding ActiveX control `AVTView` is the *master tool* of ActivView.

**Visual Basic Example**

Preparation for the following example:

- ☐ If you created a Visual Basic project for the example in the last chapter, you can continue using it.
- ☐ Otherwise, open the project `designmode\gs_designmode.vbp`. To display the form, double-click the corresponding entry in the Visual Basic Project Explorer.

The following steps are visualized in figure 3.4.

1. Add `AVTView` to the form, e.g., by double-clicking the corresponding icon in the Visual Basic toolbar with the left mouse button.

   When using Visual Basic .NET, a message box appears that warns about lost settings. This warning can be safely ignored. Adding the first ActivVisionTool to a project takes a long time, because Visual Studio .NET must perform a lot of activities: The ActivVisionTools libraries and dependencies are loaded, COM InterOp assemblies are generated, and references to those are added to the project. You see this by viewing `References` in the Solution Explorer or by checking the contents of the subdirectories `bin` and `obj` in your project directory. COM InterOp wrapper assemblies can also be generated outside the IDE using the .NET command line tools `tlbimp` and `aximp`; please consult your system's documentation for further details. Since they are generated automatically by Visual Studio .NET, we do not distribute the InterOp assemblies.

2. Perform typical design-time activities, e.g., move the control by click-and-dragging any point inside it.

3. Change some other properties in the *Property Window*, e.g., switch off the scrollbars by modifying `UseZoomScrollBars`.

4. If you are using Visual Basic .NET, the only difference is that you must add `AVTNETControl` to the form (after adding `AVTView`, otherwise an error message is displayed!) by double-clicking the icon in the toolbar.

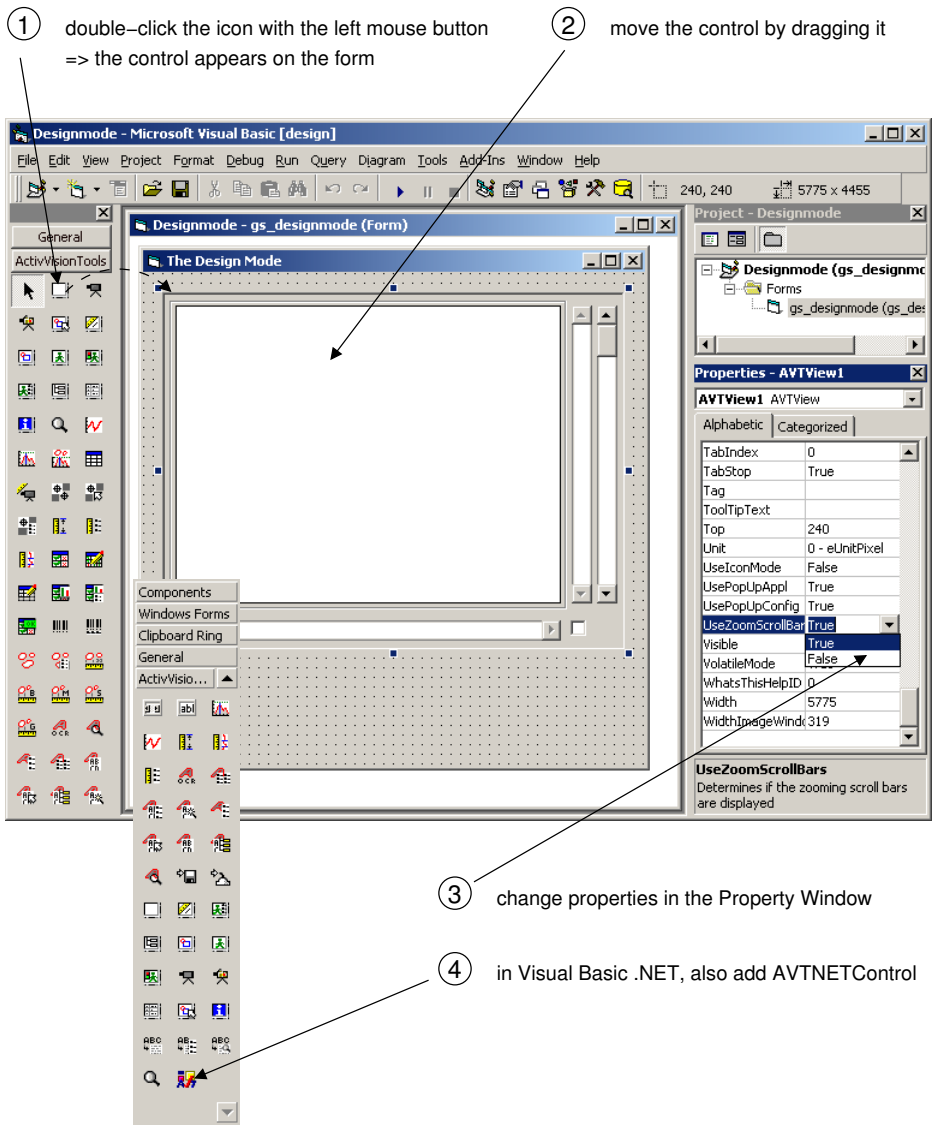
1 double–click the icon with the left mouse button
=> the control appears on the form
2 move the control by dragging it
Designmode - Microsoft Visual Basic [design]
Designmode - gs_designmode (Form)
The Design Mode
Project - Designmode
Properties - AVTView1
UseZoomScrollBars
Determines if the zooming scroll bars are displayed
3 change properties in the Property Window
4 in Visual Basic .NET, also add AVTNETControl

Figure 3.4: ActivVisionTools at design time.

## 3.3 Displaying Images

Images can be acquired via an image acquisition device or, e.g., in the testing phase, from the disk (files). Both modes are realized by the ActiveX control `AVTViewFG`, which is a *support tool* of ActivView. We start by loading images from the disk; how to connect to aimagen acquisition device is the topic of the section 3.4 on page 36.

**Visual Basic Example**

Preparation for the following example:

- ☐ If you worked on the previous example, you can continue using this project. At design time, Add `AVTViewFG` to the form by double-clicking with the left mouse button; move the appearing control to a suitable position on the form.
- ☐ Otherwise, open the project `display\gs_display.vbp`.

The following steps are visualized in figure 3.5.

1. Execute the application via the menu item `Run` ▷ `Start` or by pressing the corresponding button in the toolbar.
2. To load image files, select `'File'` in the combo box `Name`.
3. In the upper combo box in the frame `Input File`, you now can select among a list of image files provided with the ActivVisionTools CD. Alternatively, you can open a file selector by clicking `Choose File`. To open all the image files from a directory consecutively, click `Choose Directory`. This is a simplification over using a sequence file.
4. If the image is not automatically displayed, click `Single`. If you selected an image sequence, clicking this button loads the next image and displays it in the window of `AVTView`.
5. You may zoom into the image using the rightmost scroll bar of `AVTView`.
6. With the other two scroll bars, you can vary the displayed part of the image.
7. To stop the application select `Run` ▷ `Stop` or press the corresponding button in the toolbar or the exit button of the form.

Note that the image is automatically scaled to fit into the display. We recommend to adapt the size of the display so that it corresponds to the full, half, or quarter image size. How to do this is described in the User's Manual for ActivView.

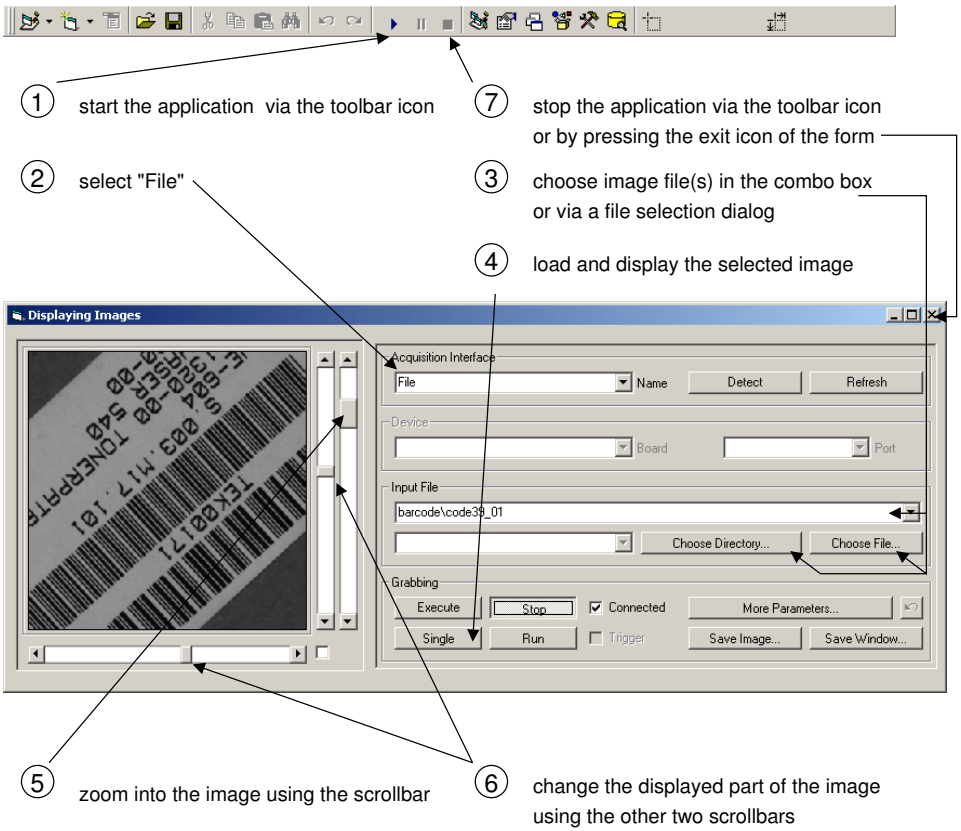


Figure 3.5: Loading images from files.

## 3.4 Connecting to an Image Acquisition Device

Before accessing an image acquisition device from within ActivVisionTools, we recommend to check that it's working correctly. Please consult the User's Manual for ActivView for more information.

**Visual Basic Example**

Preparation for the following example:

- ☐ If you worked on the previous example, you can continue using this project. Add `AVTViewStatus` to the form by double-clicking .
- ☐ Otherwise, open the project `framegrabber\gs_framegrabber.vbp`.
- ☐ Execute the application (`Run` ▷ `Start` or via the corresponding button).

The following steps are visualized in figure 3.6.

① Select your image acquisition device in the combo box `Name`.

② For some image acquisition devices, additional parameters must be specified. Please refer to the User's Manual for ActivView for more information.

③ After specifying the parameters, you can connect to the image acquisition device by checking ☑ `Connected` or clicking `Single`. This can take some time. If ActivView fails to connect to the device, a corresponding error message is displayed in `AVTView-Status`. In this case please check the User's Manual for ActivView for troubleshooting.

④ Now, you can go ahead and grab single images by clicking `Single`, grab continuously by clicking `Run`, or stop the continuous grabbing by clicking `Stop`.

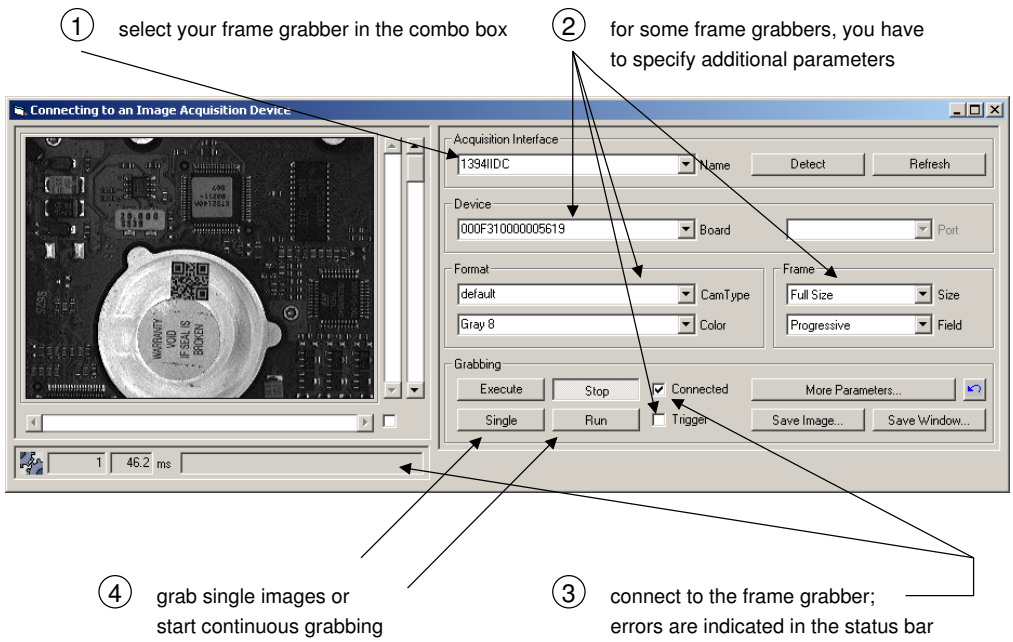


Figure 3.6: Connecting to an image acquisition device.

## 3.5 Creating a Stand-Alone Application

Up to now we have developed and run the application inside Visual Basic. To be able to run it stand-alone, i.e., without Visual Basic, you have to create an executable.

**Visual Basic Example**

Preparation for the following example:

- ☐ If you worked on the previous example, you can continue using this project.
- ☐ Otherwise, open the project `application\gs_application.vbp`.

The following steps are visualized in figure 3.7.

1. Open the dialog `File` ▷ `Make xyz.exe`; `xyz` is replaced by the name of the project, in our example `gs_application.exe`.
2. Accept the default name for the executable or choose another one. By pressing OK, the executable is created.

   Note that if the executable is created in another directory, i.e., a directory that is different from the project directory (see section 1.2.4 on page 9), this new directory will be used as the new start directory. This may lead to the loss of some settings. To further use the current settings of the project, the DSC file must be copied manually to the new directory. Therefore, we recommend to create executables only in the project directory.
3. You can start the executable from a DOS shell or from the Explorer, e.g., by double-clicking the corresponding item. It will show the same behavior as within the Visual Basic environment.

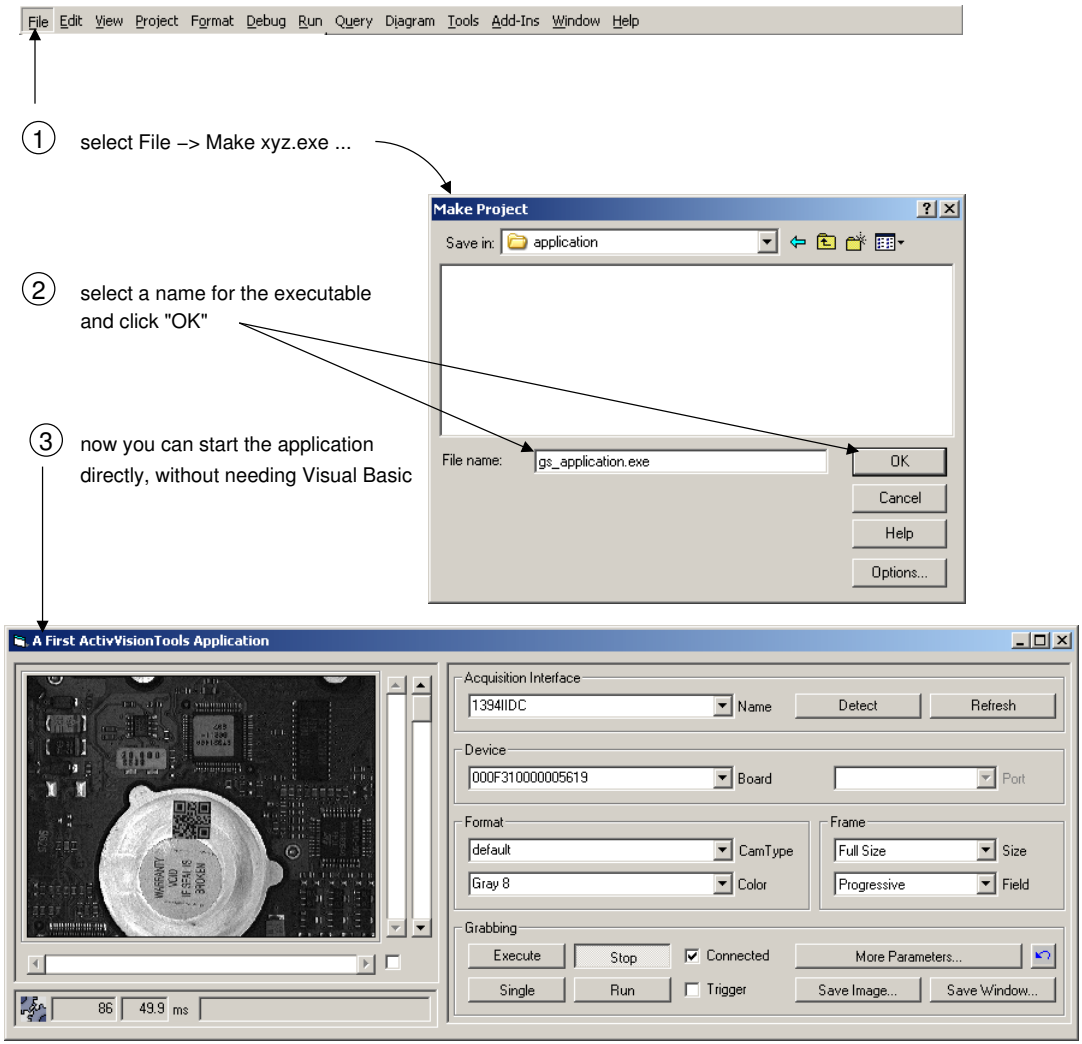


Figure 3.7: Creating a stand-alone application.

## 3.6 Checking the Necessary Licenses

In section 2.2.3 on page 20 we showed how you can check which ActivVisionTools licenses are available on your computer via ActivViewer. After developing an application, you might want to check which licenses are actually necessary for running the application. Therefore, the license information dialog can also be opened from your application via ActivView.

**Visual Basic Example**

Preparation for the following example:

- ☐ If you worked on the previous example, you can continue using this project. At design time, add `AVTMeasure` to the form by double-clicking .
- ☐ Otherwise, open the project `licensing\gs_licensing.vbp`.
- ☐ Note that if you don't have a license for ActivMeasure, an error message appears and the control remains greyed out.
- ☐ Execute the application (`Run` ▷ `Start` or via the corresponding button).

The following steps are visualized in figure 3.8.

① To open the license information dialog, click on `AVTView` with the right mouse button. In the appearing context menu select `Licenses`.

② The lower part of the dialog shows which licenses are used by the the application and whether they are available on your computer.

③ You can copy the list of used licenses to the clipboard with the button `Copy Used`.

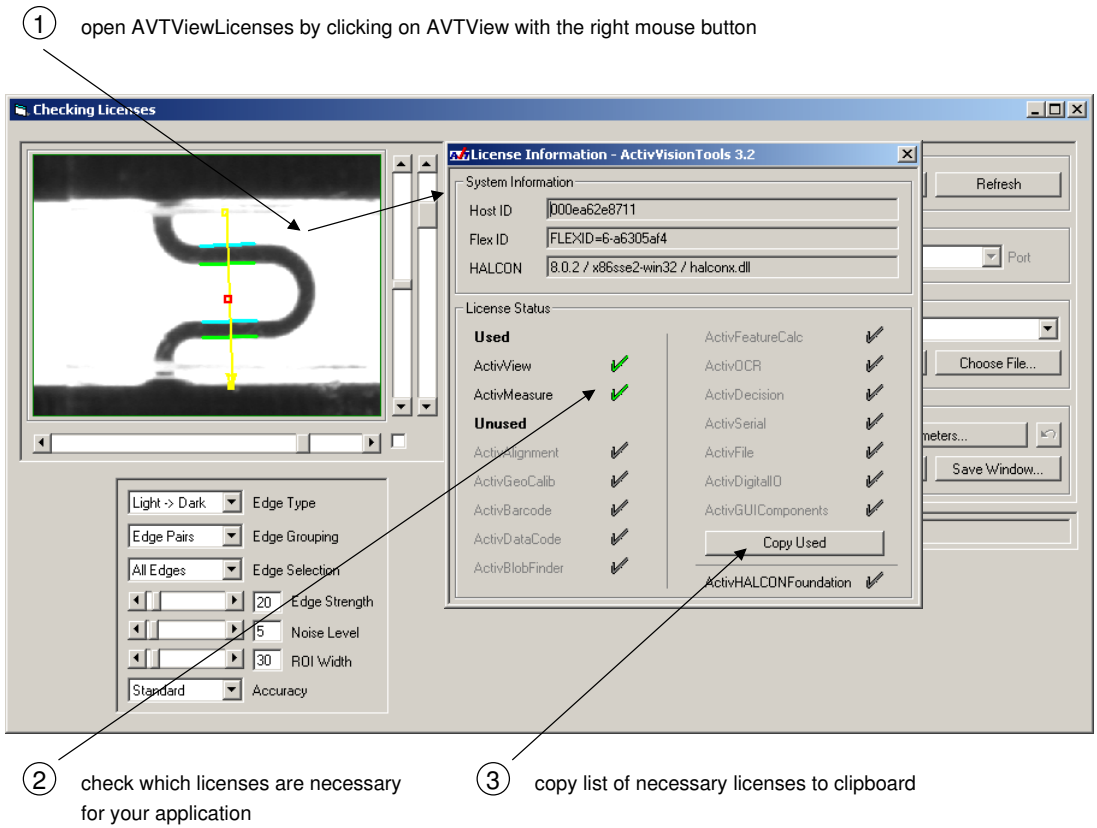


Figure 3.8: Checking which licenses are necessary to run the application.

## 3.7 Deploying an Application

To deploy an ActivVisionTools application on another computer, you need

- a valid ActivVisionTools installation and
- the run-time version of the development environment you used to create the application. For Visual Basic and Visual C++ 6.0, the required components should already be included in the ActivVisionTools installation. For .NET applications, the .NET Framework, which can be downloaded from Microsoft's web pages, is sufficient; please check that you use the correct version, e.g., 1.0 for Visual Studio .NET 2002.

Then, you copy the executable together with its DSC file and, if existing, the files containing the alignment anchor and the data code model (see the manuals User's Manual for ActivAlignment and User's Manual for ActivDataCode, respectively).

For a Visual Studio .NET application (except unmanaged C++), you simply copy the subdirectory `bin` of the release configuration of your project.

# Chapter 4

# Tips & Tricks

This chapter contains additional information which helps working with ActivVisionTools, e.g., where to find error log files or how to install the dongle driver manually. The chapter ends with a list of problems you might encounter with ActivVisionTools in general, together with tips how to solve them.

**4.1 Error Messages and the Status of an Application . . . . . 44**
**4.2 Installing the Dongle Driver Manually . . . . . 44**
4.2.1 Dongles for the Parallel Port . . . . . 44
4.2.2 Dongles for the USB Port . . . . . 45
**4.3 Troubleshooting . . . . . 46**
4.3.1 Licensing . . . . . 46
4.3.2 Using ActivVisionTools . . . . . 47
4.3.3 What to Put Into a Problem Report . . . . . 48

## 4.1 Error Messages and the Status of an Application

In an ActivVisionTools application, three classes of errors may occur: Run-time errors which are not critical to the application are logged in files for a later review. Critical errors, e.g., the failure to connect to an image acquisition device, are indicated in `AVTViewStatus`, a support tool of ActivView. This tool also displays the status of the application (see the User's Manual for ActivView). Finally, in case of a severe error like for example a missing license an error dialog pops up.

Non-critical run-time errors are logged in the file `ActivVisionTools.log`, which resides in the directory specified in the environment variable `TEMP`. Please attach this file when sending a problem report to your distributor (see also section 4.3.3 on page 48).

Note that error messages are appended to the log file until it reaches a size of 20kByte; then, at the next start of an ActivVisionTools application the log file will be cleared.

## 4.2 Installing the Dongle Driver Manually

If you did not select the dongle drivers to be installed during the installation of ActivVisionTools, you can install it manually afterwards as described in the following two sections, which correspond to the two types of supported dongles: dongles for the parallel port and USB dongles. Please note however, that you cannot use any dongle but only those supplied by MVTec via your distributor.

### 4.2.1 Dongles for the Parallel Port

You can install, configure, and uninstall the dongle driver manually using an auxiliary program which is part of each ActivVisionTools installation. Please note that **you need administrator privileges to install the driver!** If you try to install the driver without administrator privileges, the setup program might falsely state that the installation was successful.

1. Open a Windows Command Prompt and change into the folder where you installed HALCON (typically `C:\Program Files\MVTec\Halcon` , see section 2.1.1 on page 16). From there, change into the subdirectory `FLEXlm\x86-win32\flexid6-7\win_nt`. Here, start the program `setupx86.exe`; a large window appears.

   As an alternative, open the dialog `Start` ▷ `Run` and start the program by specifying its full path.

2. Select the menu item `Functions` ▷ `Install` which starts the installation process.

3. Afterwards, reboot your computer.

To check whether the dongle driver was installed successfully, open the Windows dialog Start ▷ Settings ▷ Control Panel ▷ Devices. Now, there should be an entry called Sentinel.

The dongle driver setup program can also be used to configure the driver, e.g., if your computer has more than one parallel port.

### 4.2.2 Dongles for the USB Port

Depending on your operating system, you can install, configure, and uninstall the dongle driver manually using an auxiliary program which is part of each ActivVisionTools installation. Please note that **you need administrator privileges to install the driver!**

1. Open a Windows Command Prompt and change into the folder where you installed HALCON (typically C:\Program Files\MVTec\Halcon , see section 2.1.1 on page 16). From there, change into the subdirectory FLEXlm\x86-win32\flexid9. Here, start the program hinstall.exe.

   As an alternative, open the dialog Start ▷ Run and start the program by specifying its full path.

2. This program can be used in the following ways:

```
hinstall                opens a dialog explaining how to use hinstall
hinstall -info          shows the status of the drivers
hinstall -install       installs the drivers
hinstall -remove        removes the drivers
```

3. Afterwards, reboot your computer.

To check whether the dongle driver was installed successfully, open the Windows dialog Start ▷ Settings ▷ Control Panel ▷ Devices. Now, there should be two entries called Hardlock and HASP (or similar).

## 4.3 Troubleshooting

The sections below contain information to help you to solve problems arising with ActivVision-Tools in general.

If the information given does not solve your problem, please have a look at the ActivVisionTools FAQ which can be found under `http://www.activ-vision-tools.com`. If your question is still not answered, send a problem report by email to your local distributor, not MVTec directly. See section 4.3.3 on page 48 for more information.

### 4.3.1 Licensing

If you encounter problems with your ActivVisionTools license even though your license file exists and is located in the correct directory with the correct name, a first step is always to check if the information identifying your computer (or dongle) matches the corresponding entries in the license file (see section 2.2.3 on page 20). If the two do not match, please send the new identifying information to your distributor. See below if you encounter problems with extracting the identifying information.

**?** **Licensing via network card does not work when network is not connected**

On Windows 2000 and Windows XP systems, you must disable Media Sense (DHCP) when no network is connected to the network card. Usually, it is sufficient to disable the DHCP protocol in this case. Further details can be found under `http://support.microsoft.com/support/kb/articles/Q239/9/24.asp`.

**?** **ActivVisionTools fails to detect identifiers for licensing**

If no identifiers show up in the license information dialog (see section 2.2.3 on page 20 or section 3.6 on page 40) although your computer does have a network board, a Pentium III, or a dongle, please try to extract the host IDs manually as described in the following to gain more information about the cause of trouble.

1. Open a Windows Command Prompt.
2. You can extract licensing identifiers by using the program `lmhostid` which is shipped together with the license manager FLEXlm. Just type `lmhostid` in the Command Prompt; the output might look like this:

```
> lmhostid
lmhostid - Copyright (c) 1989-2005 Macrovision Europe Ltd. and/or Macrovision
Corporation. All Rights Reserved.
The FLEXlm host ID of this machine is "00e02958e36a"
```

**Licensing via a Network Board.** If you call `lmhostid` without arguments as

above, you get the identifier corresponding to a network board. If "ffffffff" is returned, this usually indicates that you do not have a network board. If you do have one, on Windows 2000 and Windows XP systems check whether the client for Microsoft networks is installed in the dialog Start ▷ Settings ▷ Network and Dial-up Connections ▷ Local Area Connections ▷ Properties. If the client for Microsoft networks does not appear in the list install it by clicking Install and then selecting it from the list of clients. Then, call lmhostid again.

On Windows NT systems, check whether the NetBEUI and TCP/IP protocols are installed in the dialog Start ▷ Settings ▷ Control Panel ▷ Network ▷ Protocol. If the protocols do not appear in the dialog, please install these two protocols by adding them in the dialog. Then, call lmhostid again.

**Licensing via a dongle.** To get the identifier for a dongle type

```
> lmhostid -flexid
```

If this does not return the ID that is printed on the back of the dongle, check whether the dongle driver is installed and, if not, install it manually as described in section 4.2 on page 44.

If the driver is installed but lmhostid still does not return the correct ID, please check the corresponding port (parallel or USB) of your computer, before requesting a new dongle.

### 4.3.2 Using ActivVisionTools

**?** **Old Visual Basic projects cannot be loaded after upgrading ActivVisionTools**

When after upgrading ActivVisionTools you open a Visual Basic project which was created using the previous version, an error dialog pops up stating that ActivVTools.ocx could not be loaded. You can "upgrade" the project with the following method:

1. In the error popup, select Yes to proceed with the loading.
2. Another dialog pops up, stating errors during the load. Click OK to continue.
3. Open the Visual Basic menu Project ▷ Components and select ActivVisionTools 3.2 (see section 3.1 on page 26), then save the project and exit without any further action. When you open the project again, no error message should appear anymore.

Likewise, any executable created using an older version will not work anymore. Proceed similarly to above: Open the corresponding Visual Basic project, select ActivVisionTools 3.2 in the menu Project ▷ Components, save the project and exit. Open the project again and create a new executable.

**?** **An application shows strange problems which did not appear before**

In some cases the application's *description file* (see section 1.2.4 on page 9 is corrupted, e.g., if different tools are added and removed often. To test this, close Visual Basic (or the executable), remove or rename the description file, and start the application again. To check whether you found the actually used description file, take a look at its modification date; this should be the time when you last stopped the application.

### 4.3.3 What to Put Into a Problem Report

We recommend to include the following information when sending a problem report to your distributor:

- A description of your system environment, i.e., the used ActivVisionTools version, the version of the used operating system, etc.
- The exact ActivVisionTools error message which appears in `AVTViewStatus` as described in the User's Manual for ActivView. It may be easier to create a screen dump of the application and attach it to the problem report.
- The error log file described in section 4.1 on page 44.
- If the problem concerns the access to your image acquisition device, open `AVTViewFG-Parameter` (see the User's Manual for ActivView for more information) and note the content of the text box `Interface DLL`.
- If possible, attach the application itself.
- If it is not possible to send the application, please describe the steps one should take to repeat your problem. If extra data, e.g., an image, is needed to cause your problem, please attach this data to your problem report or indicate how to obtain it. Please also include any additional important information. Include all the details that would be necessary for someone else to recreate the problem, however obvious. Sometimes seemingly arbitrary or obvious information can point the way toward a solution.
- Please indicate whether you encountered any problems during the installation of Activ-VisionTools and whether the example application ActivViewer works.